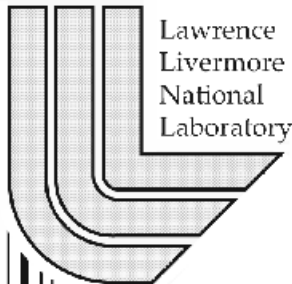


Reversible n -Bit to n -Bit Integer Haar-Like Transforms

*Joshua Senecal
Mark Duchaineau
Kenneth I. Joy*

This paper will appear in the proceedings of the 2004 Data Compression Conference, to be held at Snowbird, Utah, March 23–25, 2004

U.S. Department of Energy



Lawrence
Livermore
National
Laboratory

January 9, 2004

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Reversible n -Bit to n -Bit Integer Haar-Like Transforms

Joshua Senecal^{*‡} Mark Duchaineau[†] Kenneth I. Joy[‡]

^{*}Institute for Scientific Computing Research

[†]Center for Applied Scientific Computing
Lawrence Livermore National Laboratory

[‡]Center for Image Processing and Integrated Computing
Computer Science Department
University of California, Davis

We are researching methods of producing reversible n -bit to n -bit integer transforms. Such methods are particularly suited for hardware-based implementations, as keeping the coefficients to n bits simplifies the design of custom hardware and makes it easier to use these approaches on preexisting hardware with limited channel width.

One of our methods, called Table-Lookup Haar (TLHaar) is an approximation of the Haar Integer Wavelet Transform (Haar IWT). The Haar IWT takes two integer data values A and B and using averaging and differencing produces a low-pass value L and a high-pass value H , both integers. Because of the differencing that occurs a sign bit must be stored for nonzero H . TLHaar eliminates the need for sign bits. When operating on n -bit data, this technique uses a set of 2D lookup tables called AB2HL and HL2AB. Each of these tables is square, with an edge dimension of 2^n , and an entry width of $2n$ bits. Given two low-pass values A and B , during the forward transform A and B are used as indices into AB2HL. The entry at AB2HL[A][B] gives the corresponding high-pass coefficient H and low-pass value L . HL2AB is used when reversing the transform, taking H and L as indices and returning A and B .

If \tilde{H} and \tilde{L} indicate values produced by the original Haar transform, the lookup tables used in TLHaar have the following properties:

$$|\tilde{H}_i| \leq |\tilde{H}_j| \iff |H_i| \leq |H_j| \quad (1)$$

$$|\tilde{L}_i| \leq |\tilde{L}_j| \iff |L_i| \leq |L_j|. \quad (2)$$

That is, for any given two pairs of data values their high- and low-pass values as created by TLHaar will have the same relationships as those created by the Haar transform. To create the lookup tables we initialize each with an identity transform, giving us a 1:1 mapping between the two tables. We then sort the tables according to equations 1 and 2, until convergence.

We compared TLHaar to the Haar IWT using a variety of 8-bit image types, and performed simple compression tests. TLHaar executes up to 44% faster. Compression results are mixed, and depend on the compression method used and the image type. Future work involves using TLHaar with more sophisticated compression techniques, such as the Zerotree encoding scheme.

^{*}L-419, PO Box 808, Livermore, CA 94551, Tel: 925-422-3764, senecal1@llnl.gov

[†]duchaine@llnl.gov

[‡]kjoy@ucdavis.edu